Model Agnostic Local Explanations of Reject

André Artelt*†, Roel Visser and Barbara Hammer ‡

CITEC – Cognitive Interaction Technology
Bielefeld University – Faculty of Technology
Inspiration 1, 33619 Bielefeld – Germany

Abstract. The application of machine learning based decision making systems in safety critical areas requires reliable high certainty predictions.

Reject options are a common way of ensuring a sufficiently high certainty of predictions made by the system. While being able to reject uncertain samples is important, it is also of importance to be able to explain why a particular sample was rejected. However, explaining general reject options is still an open problem.

We propose a model agnostic method for locally explaining arbitrary reject options by means of interpretable models and counterfactual explanations.

1 Introduction

Nowadays, machine learning (ML) based decision making system are omnipresent – in particular, they are used in safety critical scenarios such as autonomous driving [1], credit (risk) assessment [2] and predictive policing [3]. Trust and reliability are critical aspects of such decision making systems.

Trust can be realized by transparency – i.e. it is difficult to trust a system that we do no not understand. It is common to achieve transparency by means of explanations – i.e. providing explanations of the systems behavior [4]. There exist different explanation methods [4] such as feature relevance/importance methods and examples based methods such as contrasting explanations.

Reliability means that we require the system to consistently output high quality predictions. However, because the models are build to output a prediction for every possible input (no matter how plausible or implausible this might be), a high quality prediction can not always be guaranteed. In particular, the certainty of the prediction might vary a lot between different inputs. Uncertain predictions are problematic in scenarios where making mistakes can have serious consequences – in such cases it might be better to refuse a prediction instead of making a potentially wrong prediction. For instance consider the example of a spam and phishing mail filter: *Imagine a mail filter application that tries to filter our spam and phishing mails in order to protect the end users and their surrounding from serious consequences. The filter is supposed to automatically sort out mails where it it is certain that the particular mails are malicious, and*

*Corresponding author: aartelt@techfak.uni-bielefeld.de
†Affiliation with the University of Cyprus

‡We gratefully acknowledge fundings from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for grant TRR 318/1 2021 - 438445824, and the VW-Foundation for the project *IMPACT* funded in the frame of the funding line *AI and its Implications for Future Society*.

pass all benign mails to the user without any delay. However, in cases where the filter is not absolute certain about its prediction (distinguishing benign vs. malicious), it should reject this mail and pass it to a human for manually checking its content – rejected mails might be passed to the user with an additional warning of taking care or to the it-security department of the company for further investigations and improvement of the filtering application. In order to understand the rejection and to support the further development of the filtering application, it is helpful to get an explanation why the filter was not able to classify the given mail.

Related Work and our Contributions Surprisingly, there does not exist a lot of work on explaining reject options. The only work we are aware of [5], which deals with reject options for learning vector quantization (LVQ) models. However, their proposed method is completely tailored towards LVQ models and its specific reject options – i.e. it is not applicable to any other models or reject options.

In this work, we propose a model agnostic method for locally explaining any reject option – i.e. we propose a method that is applicable to any model and every possible reject option. Instead of globally explaining the given reject option, we aim for a local explanation – i.e. explaining why a particular sample was rejected or not.

The remainder of this work is structured as follows: We first briefly review the necessary foundations in Section 2 and then propose our model agnostic local explanation of reject options in Section 3. Subsequently, we empirically evaluate our proposed methods from Section 3 in Section 4. The work finishes with a summary and conclusion in Section 5.

2 Foundations

In the following, we briefly review the necessary foundations of this work. First, we introduce the general modeling of reject options in Section 2.1 and subsequently conformal prediction as a potential implementation of a reject option. Then, we briefly touch upon eXplanaible AI and discuss local approximations (see Section 2.2.1) and counterfactual explanations (see Section 2.2.2).

2.1 Reject Options

Given an arbitrary classifier $h : \mathcal{X} \to \mathcal{Y}$, a reject option [6] is usually added by providing an additional function $r_h : \mathcal{X} \to \mathbb{R}_+$ that measures the certainty of classifying $\vec{x}$ and rejects a sample $\vec{x}$ if the certainty is below a given threshold θ:

$$r_h(\vec{x}) < \theta \tag{1}$$

where the subscript h denotes a potential dependency on the classifier $h(\cdot)$.

We can think about enriching a classifier $h(\cdot)$ with a reject option as constructing a new classifier $h' : \mathcal{X} \to \mathcal{Y} \cup \{\infty\}$ where we add a reject symbol ∞ to

the set of possible predictions $\mathcal{Y}$:

$$h'(\vec{x}) = \begin{cases} h(\vec{x}) & \text{if } r_h(\vec{x}) \geq \theta \\ \infty & \text{otherwise} \end{cases} \tag{2}$$

In the following, we briefly introduce conformal prediction as a specific way of realizing such a reject option $r(\cdot)$.

2.1.1 Conformal Prediction for Implementing a Reject Option

Assume that a (black-box) probabilistic classifier $h : \mathcal{X} \to \mathcal{Y}$ of the following form is given:

$$h(\vec{x}) = \underset{y \in \mathcal{Y}}{\arg\max}\, p(y \mid \vec{x}) \tag{3}$$

where $p(y \mid \vec{x})$ denotes the class wise probability as estimated by the classifier $h(\cdot)$.

A central building block of a conformal predictor [7] is a so called non-conformity measure $\phi_h : \mathcal{X}, \mathcal{Y} \to \mathbb{R}$ which measures how different a given labeled sample is from a given set of labeled samples we have seen before. In case of a probabilistic classifier $h(\cdot)$, a common non-conformity measure is given as follows:

$$\phi_h(\vec{x}, y = j) = \max_{i \neq j} p_h(y = i \mid \vec{x}) - p_h(y = j \mid \vec{x}) \tag{4}$$

For calibrating (fitting) a conformal predictor based on $h(\cdot)$, we need another labeled data set $\mathcal{D}_{\text{calib}} \subset \mathcal{X} \times \mathcal{Y}$ which was not used during the fitting of $h(\cdot)$. Calibrating/fitting an (inductive) conformal predictor means to compute the non-conformity α_i of every sample from the calibration set by applying $\phi_h(\cdot)$:

$$\alpha_i = \phi_h(\vec{x}_i, y = y_i) \tag{5}$$

For every new data point $\vec{x}_* \in \mathcal{X}$ that is going to be classified, we compute the non-conformity measure for every possible label in $\mathcal{Y}$:

$$\alpha_*^i = \phi_h(\vec{x}_*, y = i) \tag{6}$$

Next, the non-conformity scores of $\vec{x}_*$ are compared with the non-conformity scores from the calibration set to compute p-values for every possible classification of $\vec{x}_*$:

$$p_{y=i}(\vec{x}_*) = \frac{|\alpha_j \in \mathcal{D}_{\text{calib}} \geq \alpha_*^i|}{|\mathcal{D}_{\text{calib}}| + 1} \tag{7}$$

The conformal predictor then selects the label with the larges p-value as a prediction – i.e. Eq. (3) becomes:

$$h(\vec{x}_*) = \underset{i \in \mathcal{Y}}{\arg\max}\, p_{y=i}(\vec{x}_*) \tag{8}$$

The confidence of the prediction – i.e. how likely (given the training set) the prediction is compared to all other possible predictions – is then computed as follows:

$$1 - \left(\max_{j \neq \arg\max_i p_{y=i}(\vec{x}_*)} p_{y=j}(\vec{x}_*) \right) \tag{9}$$

and the credibility – i.e. how well the training set supports the prediction – is given as follows:

$$\psi(\vec{x}_*) = \max_i p_{y=i}(\vec{x}_*) \tag{10}$$

In order to use conformal prediction for implementing a reject option Eq. (1) [8], one could use either the conformity score Eq. (9) or the credibility score Eq. (10). As it is common practice, we use the credibility as a reject score in this work:

$$r_h(\vec{x}) = \psi(\vec{x}) \tag{11}$$

2.2 Explanations

In the following, we briefly review two popular types of explanations. Explanations using local approximations (Section 2.2.1) and counterfactual explanations as an instance of example based explanations (Section 2.2.2).

2.2.1 Local Approximations

There exist popular methods for locally explaining a given model $h(\cdot)$, instead of trying to come up with a global explanation [4]. A common approach for local explanations is to build a local approximation of the model $h(\cdot)$ which is then used for creating an explanation.

A popular instance of such methods is LIME [9]. Here, the authors propose to fit an interpretable model (e.g. a linear model) to a set of labeled perturbed samples – i.e. the original model $h(\cdot)$ is applied to perturbed instances of the given original sample $\vec{x}_{\text{orig}}$ for which we want to compute a local explanation. The final, local, explanation is then constructed using the most relevant features of the local approximation – in order to get a meaningful explanation, the features must be interpretable and meaningful (e.g. super-pixels in case of images).

Another method that uses local approximations for computing a local explanation is Anchors [10]. Anchors are if-then rules based explanations that locally explain the prediction of the given model $h(\cdot)$.

2.2.2 Counterfactual Explanations

Counterfactual explanations (often just called *counterfactuals*) are a prominent instance of contrasting explanations, which state a change to some features of a given input such that the resulting data point, called the counterfactual, causes a different behavior of the system than the original input does. Thus, one can think of a counterfactual explanation as a suggestion of actions that change the

model's behavior/prediction. One reason why counterfactual explanations are so popular is that there exists evidence that explanations used by humans are often contrasting in nature [11] – i.e. people often ask questions like *"What would have to be different in order to observe a different outcome?"*. Despite their popularity, the missing uniqueness of counterfactuals could pose a problem: Often there exist more than one possible/valid counterfactual – this is called the Rashomon effect [4] – and in such cases, it is not clear which or how many of them should be presented to the user. One common modeling approach (if this problem is not simply ignored) is to enforce uniqueness by a suitable formalization.

In order to keep the explanation (suggested changes) simple – i.e. easy to understand – an obvious strategy is to look for a small number of changes so that the resulting sample (counterfactual) is similar/close to the original sample, which is aimed to be captured by Definition 1.

Definition 1 ((Closest) Counterfactual Explanation [12])**.** *Assume a prediction function $h : \mathbb{R}^d \to \mathcal{Y}$ is given. Computing a counterfactual $\vec{x}_{cf} \in \mathbb{R}^d$ for a given input $\vec{x}_{orig} \in \mathbb{R}^d$ is phrased as an optimization problem:*

$$\underset{\vec{x}_{cf} \in \mathbb{R}^d}{\arg\min} \; \ell\left(h(\vec{x}_{cf}), y'\right) + C \cdot \theta(\vec{x}_{cf}, \vec{x}_{orig}) \tag{12}$$

where $\ell(\cdot)$ denotes a loss function, y' the target prediction, $\theta(\cdot)$ a penalty for dissimilarity of $\vec{x}_{cf}$ and $\vec{x}_{orig}$, and $C > 0$ denotes the regularization strength.

In the following, we assume a binary classification problem: In this case, we denote a (closest) counterfactual $\vec{x}_{\text{cf}}$ according to Definition 1 of a given sample $\vec{x}_{\text{orig}}$ under a prediction function $h(\cdot)$ simply as $\vec{x}_{\text{cf}} = \text{CF}(\vec{x}_{\text{orig}}, h)$ and drop the target label y' because it is uniquely determined.

The counterfactuals from Definition 1 are also called *closest counterfactuals* because the optimization problem Eq. (12) tries to find an explanation $\vec{x}_{\text{cf}}$ that is as close as possible to the original sample $\vec{x}_{\text{orig}}$. However, other aspects like plausibility and actionability are ignored in Definition 1, but are covered in other work [13, 14, 15] – note that it is not always clear which type of counterfactual is meant when people talk about counterfactuals. In this work, we use the term counterfactuals in the spirit of Definition 1.

3 Local Approximations for Explaining Reject

We propose a model agnostic approach for locally explaining arbitrary reject options – i.e. our method does not need access to the reject option or the underlying ML model, access to a prediction interface is sufficient. Instead of explaining the reject option globally, we aim for a local explanation only – i.e. explaining the reject of a particular sample.

Given a sample $\vec{x}_{\text{orig}} \in \mathcal{X}$ which is rejected by the reject option, we sample a fixed number of samples $\{\vec{x}_i\}$ from the neighborhood around $\vec{x}_{\text{orig}}$ and label

each sample whether it is also rejected or not:

$$y_i = \begin{cases} 1 & \text{if } r(\vec{x}_i) < \theta \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, \vec{x}_i \in \mathcal{B}_\epsilon(\vec{x}_{\text{orig}}) \tag{13}$$

where $\mathcal{B}_\epsilon(\vec{x}_{\text{orig}})$ denotes a fixed number of samples in the neighborhood of $\vec{x}_{\text{orig}}$. We then fit an interpretable classifier h_{local} (e.g. a linear model or a decision tree) to these samples $\mathcal{D}_{\text{local}} = \{(\vec{x}_i, y_i)\}$.

We propose to either use $h_{\text{local}}(\cdot)$ as an explanation – e.g. using the obtained feature importances or learned decision rules as an explanation –, or a counterfactual explanation (see Definition 1) $\vec{x}_{\text{cf}} = \text{CF}(\vec{x}_{\text{orig}}, h_{\text{local}})$ of $h_{\text{local}}(\cdot)$ as an explanation of the reject of $\vec{x}_{\text{orig}}$.

Formally, we propose two different realizations of a local explanation Ψ at $\vec{x}_{\text{orig}}$ under a given reject option $r(\cdot)$:

$$\Psi(r, \vec{x}_{\text{orig}}) = \begin{cases} \text{FRI}(h_{\text{local}}) \\ \text{CF}(\vec{x}_{\text{orig}}, h_{\text{local}}) \end{cases} \tag{14}$$

where $\text{FRI}(\cdot)$ denotes the feature relevance as obtained from a given model. In the experiments (Section 4), we empirically evaluate and compare both types of explanations in the experiments (Section 4).

4 Experiments

In the following, we empirically evaluate our proposed model agnostic methods for explaining rejects (see Section 3). We do so by considering two different aspects for evaluation:

- We evaluate computational aspects like sparsity of the computed explanations.
- We evaluate the ground truth recovery rate (goodness) of the explanations by evaluating if and how well the explanations match the ground truth – i.e. identifying the relevant features.

In addition, we always evaluate the accuracy of the learned local approximation – i.e. checking if the original sample is also rejected under the local approximation. All experiments are implemented in Python and the implementation is publicly available on GitHub[1].

4.1 Data Sets

We consider the following data sets for our empirical evaluation – all data sets are scaled and standardized:

[1] `https://github.com/andreArtelt/LocalModelAgnosticExplanationReject`

4.1.1 Wine

The "Wine data set" [16] is used for predicting the cultivator of given wine samples based on their chemical properties. The data set contains 178 samples and 13 numerical features such as alcohol and color intensity.

4.1.2 Breast cancer

The "Breast Cancer Wisconsin (Diagnostic) Data Set" [17] is used for classifying breast cancer samples into benign and malignant. The data set contains 569 samples and 30 numerical features such as smoothness and compactness.

4.1.3 Flip

This data set [18] is used for the prediction of fibrosis. The set consists of samples of 118 patients and 12 numerical features such as blood glucose, BMI and total cholesterol. As the data set contains some rows with missing values, we chose to replace these missing values with the corresponding feature mean.

4.1.4 t21

This data set [19] is used for early diagnosis of chromosomal abnormalities, such as trisomy 21, in pregnant women. The data set consists of 18 numerical features such as heart rate and weight, and contains over 50000 samples but only 0.8 percent abnormal samples (e.g. cases of trisomy 21) – i.e. it is highly imbalanced.

4.2 Setup

Since our method (see Section 3) is completely model agnostic, we evaluate it on a set of diverse classifiers: k-nearest neighbors classifier (kNN), Gaussian naive Bayes classifier (GNB), random forest classifier (RandomForest). Whereby we always use conformal prediction (see Section 2.1) for realizing a credibility based reject option Eq. (11).

In order to make a fair comparison between the efficacy of the methods we perform hyperparameter tuning in order to find the best performing model parameters. This includes the hyperparameters of the respective classifiers, which are obtained by a grid search on each of them. Additionally, we try to find an appropriate rejection threshold by using the Knee/Elbow method [20] for finding a reasonable cut-off point. Using the Kneedle algorithm [20], we can find an appropriate rejection threshold by determining the "optimal" threshold in the ARC by finding the so-called knee-point. In a real world scenario the threshold might be tuned to allow for a more relaxed or strict rejection scenario, however for the purpose of our research finding the knee point gives us a fairly good approximation of what would usually be considered an appropriate or well performing rejection threshold.

We run all experiments in a 5-fold cross validation to get meaningful and statistically reliable results – we consider every possible combination of data set and classifier. We use a decision tree classifier for the computation of the local approximation. After fitting the classifier, we apply the reject option to all samples from the test set and compute explanations for those that are rejected by the reject option. As proposed in Section 3, we always compute two explanations: feature relevance profile as obtained from the local approximation – we use the Gini importance as obtained from the decision tree classifier – and a counterfactual explanation under this local approximation.

Algorithmic Properties When evaluating algorithmic properties, we not only compute the accuracy – i.e. is the prediction of the local approximation consistent with the prediction of the original model –, but also compute the sparsity (l_0-norm) of both explanations.

Goodness of Explanations For evaluating the goodness of the explanations, we create scenarios with known ground truth as follows: For each data set, we select a random subset of features (30%) and perturb these in the test set by adding Gaussian noise – we then check which of these samples are rejected due to the noise (i.e. applying the reject option before and after applying the perturbation), and compute explanations of these samples only. Finally, we evaluate for both explanations how many of the relevant features (from the known ground truth) are recovered and included in the explanation.

4.3 Results & Discussion

When reporting the results, we use the following abbreviations: *FeatImp* – Feature importances as obtained from the local approximation; *Cf* – Counterfatual explanation. Note that we round all values to two decimal points.

Algorithmic Properties We report the mean accuracy and sparsity in Table 1. We observe that the local approximation is usually sufficiently good (although some combinations of model and data set seem to be more challenging) and the final explanations are very sparse – i.e. we obtain low-complexity explanations. Furthermore, we observe that counterfactual explanations of the local approximation are consistently sparser than the obtained feature importance.

Goodness of Explanations The mean recall of correctly recovered relevant features is given in Table 2. First, we observe that the perturbation does not strongly affect the accuracy. Next, we observe that both explanations have trouble to recover all perturbed features – although the feature importance explanation recovers consistently more perturbed features than the counterfactual explanations. The reasons for this are two-fold: First, because we optimized sparsity (i.e. getting low-complexity explanations), the explanations contain very few features only and are therefore likely to miss some perturbed features.

Table 1: Algorithmic properties – Mean (incl. variance) accuracy and sparsity – larger values are "better" for accuracy, while smaller values are "better" for sparsity.

	DataSet	Accuracy	FeatImp	Cf
kNN	Wine	0.80 ± 0.16	4.5 ± 1.98	1.25 ± 0.23
	Breast Cancer	0.92 ± 0.0	5.12 ± 1.66	1.25 ± 0.19
	t21	0.96 ± 0.00	3.9 ± 3.43	1.07 ± 0.27
	Flip	0.31 ± 0.07	5.21 ± 1.13	1.00 ± 0.00
GNB	Wine	0.92 ± 0.00	4.57 ± 1.17	1.11 ± 0.10
	Breast Cancer	0.88 ± 0.00	3.83 ± 1.38	1.07 ± 0.07
	t21	0.78 ± 0.15	1.12 ± 1.71	0.71 ± 0.26
	Flip	0.83 ± 0.01	1.73 ± 0.54	1.00 ± 0.00
RandomForest	Wine	0.8 ± 0.16	3.26 ± 1.64	1.43 ± 0.37
	Breast Cancer	1.00 ± 0.00	1.07 ± 2.35	0.52 ± 0.48
	t21	0.95 ± 0.00	3.75 ± 2.59	1.22 ± 0.30
	Flip	0.50 ± 0.06	5.05 ± 1.33	1.05 ± 0.05

Table 2: Goodness of explanations – Mean (incl. variance) recall of correctly identified relevant features (larger numbers are better).

	DataSet	Accuracy	FeatImp	Cf
kNN	Wine	0.75 ± 0.15	0.53 ± 0.03	0.28 ± 0.15
	Breast Cancer	0.89 ± 0.02	0.50 ± 0.04	0.23 ± 0.12
	t21	0.78 ± 0.02	0.56 ± 0.03	0.36 ± 0.15
	Flip	0.40 ± 0.14	0.30 ± 0.08	0.04 ± 0.03
GNB	Wine	0.85 ± 0.04	0.56 ± 0.06	0.43 ± 0.18
	Breast Cancer	0.97 ± 0.0	0.39 ± 0.09	0.23 ± 0.12
	t21	0.60 ± 0.24	0.45 ± 0.13	0.36 ± 0.15
	Flip	0.91 ± 0.01	0.40 ± 0.14	0.38 ± 0.18
RandomForest	Wine	1.00 ± 0.00	0.51 ± 0.13	0.39 ± 0.16
	Breast Cancer	1.00 ± 0.00	0.18 ± 0.08	0.16 ± 0.09
	t21	0.62 ± 0.11	0.58 ± 0.05	0.50 ± 0.15
	Flip	0.61 ± 0.08	0.54 ± 0.08	0.38 ± 0.15

Second, it seems that the local approximation is not sensitive enough to the applied perturbations – the accuracy is pretty high, but still the explanations have trouble identifying all perturbed features.

5 Summary & Conclusion

In this work, we proposed a model agnostic approach for explaining reject options: we proposed to use a local approximation of the reject option and explain the reject locally either by the local approximation itself (assuming that this local approximation is interpretable) or by counterfactual explanations of this local approximation. We empirically evaluated these two explanations methods under computational as well as qualitative aspects. We observed reasonable performance of both explanations – in particular counterfactual explanations were able to come up with low complexity explanations but identified fewer of the relevant features.

The empirical evaluation in this work focuses on computational proxies only. However, it still remains unclear if and how useful our proposed explanations are to humans. Since it is difficult to phrase "usefullness" as a scoring function, a proper use study is needed. We leave this aspects as future work.

References

[1] Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. Electronic Imaging, 2017(19):70–76, 2017.

[2] Amir E. Khandani, Adlar J. Kim, and Andrew Lo. Consumer credit-risk models via machine-learning algorithms. Journal of Banking & Finance, 34(11), 2010.

[3] Panagiotis Stalidis, Theodoros Semertzidis, and Petros Daras. Examining deep learning architectures for crime classification and prediction. abs/1812.00602, 2018.

[4] Christoph Molnar. Interpretable Machine Learning. 2019.

[5] André Artelt, Johannes Brinkrolf, Roel Visser, and Barbara Hammer. Explaining reject options of learning vector quantization classifiers, 2022.

[6] Kilian Hendrickx, Lorenzo Perini, Dries Van der Plas, Wannes Meert, and Jesse Davis. Machine learning with a reject option: A survey. arXiv preprint arXiv:2107.11277, 2021.

[7] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. J. Mach. Learn. Res., 9:371–421, 2008.

[8] Henrik Linusson, Ulf Johansson, Henrik Boström, and Tuve Löfström. Classification with reject option using conformal prediction. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 94–105. Springer, 2018.

[9] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, editors, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, pages 1135–1144. ACM, 2016.

[10] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 1527–1535. AAAI Press, 2018.

[11] Ruth M. J. Byrne. Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. In IJCAI-19, 2019.

[12] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harv. JL & Tech., 31:841, 2017.

[13] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. CoRR, abs/1907.02584, 2019.

[14] André Artelt and Barbara Hammer. Convex density constraints for computing plausible counterfactual explanations. 29th International Conference on Artificial Neural Networks (ICANN), 2020.

[15] André Artelt and Barbara Hammer. Convex optimization for actionable \& plausible counterfactual explanations. CoRR, abs/2105.07630, 2021.

[16] D. Coomans S. Aeberhard and O. de Vel. Comparison of classifiers in high dimensional settings. Tech. Rep. no. 92-02, 1992.

[17] Olvi L. Mangasarian William H. Wolberg, W. Nick Street. Breast cancer wisconsin (diagnostic) data set. `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)`, 1995.

[18] Jan-Peter Sowa, Dominik Heider, Lars Peter Bechmann, Guido Gerken, Daniel Hoffmann, and Ali Canbay. Novel algorithm for non-invasive assessment of fibrosis in nafld. PLOS ONE, 8(4):1–6, 04 2013.

[19] K. H. Nicolaides, K. Spencer, K. Avgidou, S. Faiola, and O. Falcon. Multicenter study of first-trimester screening for trisomy 21 in 75 821 pregnancies: results and estimation of the potential impact of individual risk-orientated two-stage first-trimester screening. Ultrasound in Obstetrics & Gynecology, 25(3):221–226, 2005.

[20] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In 2011 31st International Conference on Distributed Computing Systems Workshops, pages 166–171, 2011.